Technology for Facilitating and Analyzing Interviews in Large Groups

by

Joey Y. Zhou

S.B., E.E.C.S. M.I.T., 2007

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
NOV 13 2008
LIBRARIES

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 2008

Copyright 2008 Joey Y. Zhou. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
to distribute publicly paper and electronic copies of this thesis document in whole and in part
in any medium now known or hereafter created.

Author_____
Department of Electrical Engineering and Computer Science
May 23, 2008

Certified by_____
Charles E. Leiserson
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

ARCHIVES

Technology for Facilitating and Analyzing Interviews in Large Groups

by

Joey Y. Zhou

Submitted to the

Department of Electrical Engineering and Computer Science

May 23, 2008

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

This thesis presents a system for leveraging simple technologies to improve the robustness, efficiency, and effectiveness of interviews in the context of a large network. The technology was used in the context of a successful activity—the Leadership Module—to teach leadership and communication to a group of engineering sophomores. The Leadership Module combines principles in education, organization behavior, and algorithms to improve the student learning experience. Two separate iterations of the activity explored the tradeoff between data gathering and student feedback. The power method of eigenvalue analysis was used to rank student evaluations of leadership effectively and robustly, yet provided insufficient feedback for students. Revising the activity to focus on visioning and collaborating resulted in a more relevant lesson for students. The system can be extended to other applications, especially speed dating and other academic courses.

Thesis Supervisor: Charles E. Leiserson
Title: Professor of Computer Science and Engineering

Acknowledgments

First, I would like to thank my thesis supervisor, Professor Charles E. Leiserson, for giving me the opportunity to work on the UPOP Leadership Module, first as a UROP and then as a Masters of Engineering thesis. Charles has helped me with every step of the thesis, from research to development to writing, and from him I've learned many professional, academic, and life lessons.

This thesis would not have been possible without the assistance of the Undergraduate Practice Opportunities Program (UPOP), which served as the case study for this project in 2007 and 2008. I would like to thank Chris Resto and Elizabeth Arnold, who along with Charles, taught me how to teach the Leadership Module in 2007. I would also like to thank Thomas Kochan and John Carroll for their invaluable advice in developing the 2008 Leadership Module, and Susann LuperFoy and Dori Peleg for their help in successfully running the Module. I look forward to continuing work with UPOP in the fall of 2008 and 2009.

Finally, I would like to thank my family and friends, especially those at New House 4, for their support during my time at MIT.

Contents

List of Figures

Chapter 1

Background and Motivation

1.1 Introduction

Technology has revolutionized communication in every way imaginable. Television, cellular telephones, email, and the World Wide Web have transformed the way humans interact with each other, in pleasure, business, and politics. Technology lowers the barriers to gathering information, and increases the speed and efficiency of messages being communicated.

One area which has seemed immune from the transformational effects of technology is that of interviews. The American Heritage Dictionary defines the interview as "a formal meeting in person, especially one arranged for the assessment of the qualifications of an applicant [1]." Colloquially, it describes an interaction, usually between two people, that involves the mutual exchange of information and possibly judgment. Interviews are most commonly used in the context of finding a job. They also take place in social settings: a romantic date between two people can be considered an interview.

Regardless of the situation, the basic interview has remained constant in the past few decades. Technology has enabled variations of interviews, such as phone interviews or online assessment exams. But the majority of employers prefer the original interview—there is no replacement for the face-to-face conversation between human beings. For many companies, the interview is a

central part of the hiring process, even though Malcolm Gladwell argued in 2000 that interviews are highly subjective, and they cause people to make lasting personality judgemnts based on only a snapshot of one's behavior [5].

Recently, speed dating has become a popular phenomenon with individuals seeking romantic partners. Speed dating takes advantage of the power of first impressions—studies have shown that human beings form strong opinions of each other at the moment of meeting [2][3][4]. Compared to normal dating activities, speed dating has the advantages of efficiency and honesty. Participants can meet many potential partners in a short time-frame, and people need not evaluate their partners until after meeting them.

This thesis presents a new system for interviews involving large groups of people. It uses simple, inexpensive technology to support the interview process. The interviews are studied in the context of two UPOP Leadership Modules, which will be explained in Sections 1.2 and 1.3. The main process has been developed using leadership theory, network theory, and computer science. This thesis will explore the challenges and tradeoffs that affected the first two iterations of the Leadership Module. It presents the framework and tools to evaluate large groups of candidates, with applications in speed-dating, voting, interviewing, and teaching.

1.2 The Undergraduate Practice Opportunities Program (UPOP)

The Undergraduate Practice Opportunities Program (UPOP) was started in 2001 to teach MIT sophomores how to apply their engineering knowledge to solve real-world challenges. According to the UPOP website, "UPOP is a co-curricular program that exposes talented undergraduates to the multi-faceted nature of professional practice and helps them make more effective transitions from academia to the real world." The students that participate "receive coaching from dynamic engineering and management faculty and real industry professionals as they engage in a pre-employment boot camp during an intense weeklong session in January and several evening meetings throughout the spring [20]."

The main focus of UPOP takes place in January, when two groups of students each take a week-long workshop. The aim of the workshop is to introduce students to real-world engineering

challenges, such as conflict resolution, effective communication, system dynamics, networking, and people management. The UPOP experience culminates in a summer internship, and the workshop helps prepare students for adapting from an academic to industrial setting.

UPOP provides an interesting setting for the leadership activity:

- Each group has over a hundred students.
- The students are relatively homogeneous academically, being sophomores from the MIT School of Engineering.
- The students are committed to finding an internship. Therefore, they are focused on improving their professional skills.

1.3 History of the UPOP Leadership Module

The Leadership Module is an integral part of UPOP, teaching engineering sophomores how to apply leadership in their work experiences. Previous to 2007, it consisted of students watching videos of complex engineering projects, such as the Big Dig in Boston. Concern with the enthusiasm and involvement of students led to a complete overhaul of the UPOP Leadership Module in 2007 by Charles E. Leiserson of CSAIL. The new design has two parts: the theory and presentation of leadership, and the practical application of leadership skills in the form of an interview game. Currently it is taught by Professors Tom Kochan and John Carroll at the Sloan School of Management. I helped design the theoretical component and I managed the practical exercise of the UPOP Leadership Module in 2007 and 2008. The leadership module serves as a case study that illustrates the principles of system design proposed in this paper.

1.4 Thesis Structure

Chapter 2 provides an overview of the system design used by the UPOP Leadership Module. The next three chapters cover the software engineering of the system. Much of the work centered on integrating human users with technology to organize and run the Leadership Module. Chapter 3 explains the Dance-Card Generator, a tool used to generate random pairs of leaders and followers for multiple rounds of the exercise. Chapter 4 covers the Data Collector, a combination of forms,

form reader, and software used to operate the form reader. Chapter 5 describes the Rank Calculator, a software program that compute rankings of students using eigenvalue analysis.

Chapter 6 and Chapter 7 implement the software tools in actual test cases. Chapter 6 describes the 2007 UPOP Leadership Module, which incorporates the Pair Generator, Data Collector, and Rank Calculator. The 2008 UPOP Leadership Module changed parts of the system, and is presented in Chapter 7. Chapter 8 and Chapter 9 conclude with an evaluation of the two Leadership Modules and suggest changes for future iterations. The Appendices show the development of scantron forms for the activity. Appendix A presents the forms for the 2007 Leadership Module, and Appendix B presents the forms for the 2008 Leadership Module.

Chapter 2

Overview of the UPOP Leadership Module

The activity for the Leadership Module consists of a system with three parts – the technical components, the human users, and the processes that connect the two. The objective of the system is to maximize effectiveness of the activity given limited resources. Effectiveness is defined by the lessons learned and enjoyment expressed by students. This objective is applied to all system components, seeking to generate the maximum information while maintaining simplicity and economy of supporting technology.

The theory of distributed leadership, outlined in Section 2.2, is fundamental to the overall design of the Leadership Module, which is described in Section 2.3. The chapter concludes with the theory of human fallibility—a central concern in every aspect of the Leadership Module. Through the implementation and evaluation of the system, we're continually reminded that human beings are unreliable, and a robust system must account for human error to operate successfully.

2.1 The Sloan Leadership Framework: Distributed Leadership Model

To teach leadership effectively, in 2007 Professor Leiserson and I created a system where students would define new and creative ideas in the form of MIT hacks, then share those ideas with their peers. In 2008, the system was refined with the idea of "visioning", the creation of a vision to change any aspect of the world. Both instances of the Leadership Module related to the Sloan Leadership Framework, which is described in this section. Section 6.1 covers creating hacks in more detail. Section 7.1 specifically defines the visioning activity.

The Sloan School of Management has developed a Distributed Leadership Model (DLM) [6], shown in Figure 2-1, for understanding leadership in the context of organizations and businesses. It makes the core assumptions that leadership is distributed across all departments and levels of an organization; each leader develops his or her personal style of leadership; leaders learn over time; leadership is about changing the status quo. Developed over time by Deborah Ancona, Tom Malone, Wanda Orlikowski, and Peter Senge, it defines four dimensions that are encompassed by leadership.

1. **Sense-making** is making sense of and mapping the environment. Through sense-making, the leader seeks to understand the situation or context in which he or she operates. Thus, it mainly involves intelligence and quickly processing information from a variety of sources in the environment. Sense-making also requires objectivity and the willingness to depart from accept norms.
2. **Relating** is defined by discovering, influencing, and connecting with others. The process of relating begins with the leader keeping an open mind, listening to others, and understanding their perspectives. Next, the leader must have the ability to defend opinions and advocate ideas. Finally, leaders connect with others to build professional networks, enhancing the flow of ideas and bringing people together to achieve a common goal.
3. **Visioning** is the creation of a new image of the future, with both broad ideals and concrete goals. Visions inspire people by showing them why they are working so hard. In order to create change, leaders need to have a vision of the future. Conversely, a compelling vision not only paints an inspirational picture of the future, it also convinces

the viewer of the possibility of achieving the goals to fulfill the vision. A powerful vision also requires sincerity by the leader.

4. **Inventing** is the implementation of one's vision into a reality. It involves developing a plan of action, and then defining the procedures, structures, and participants to implement the plan. Because it frequently involves creating new solutions to tackle previously unsolved problems, it requires creativity and innovation.

These four capabilities are combined with a personal change signature that defines each individual's leadership style, depending on the individual's strengths and weaknesses. The change signature is derived from one's values and experiences, and can be shown through expression of ideas, skills, and tactics.

Figure 2-1: The Sloan Distributed Leadership Framework (Ancona 2005)

From the point of view of the students, the activity in the 2008 Leadership Module is analogous to the Sloan Leadership Model, as shown in Figure 2-2. Students first create a vision for the future, defining a particular goal or wish with social impact. During the exercise, they try to convince other students to join their visions. In the context, they are relating to other people and making sense of how their visions fit in their environments. Finally, at the end of the exercise,

students are encouraged to implement their visions and take the first steps to turning visions into realities.

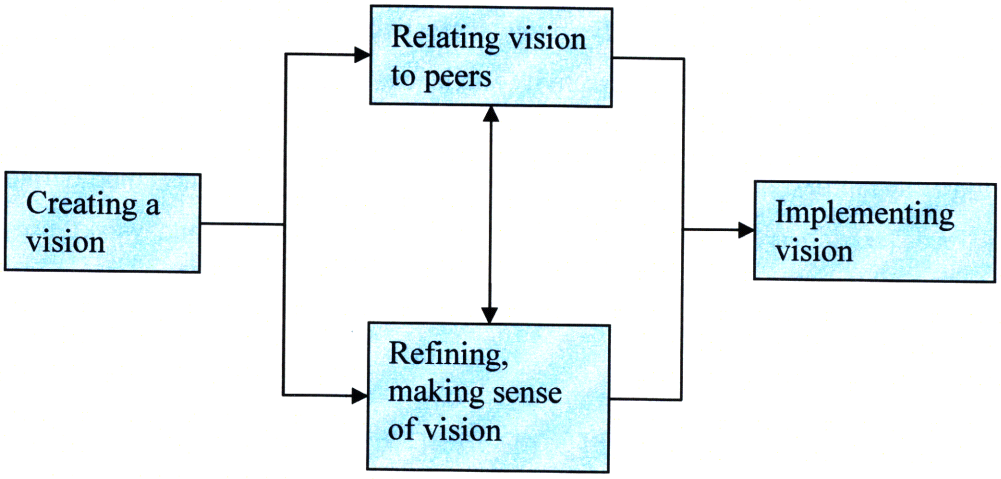


Figure 2-2: Using the Sloan Leadership Framework to describe the Leadership Module Activity

2.2 System Design

Figure 2-3 describes the concepts in the overall process. First, the students create an idea—either an MIT hack or a broad vision. Then they share that idea with their peers in an interview exercise. Due to the complexity of organizing over a hundred students, the instructions for the exercise are detailed and essential to a successful execution. In the exercise, each student participates in a number of rounds, with a different partner in each round. Each round is structured so that students first find partners, then interview their partners, and finally evaluate their partners. After the exercise, I use the software to provide feedback for the students based on their evaluations. Chapter 6 and Chapter 7 explain the detailed execution of the system in 2007 and 2008 respectively.

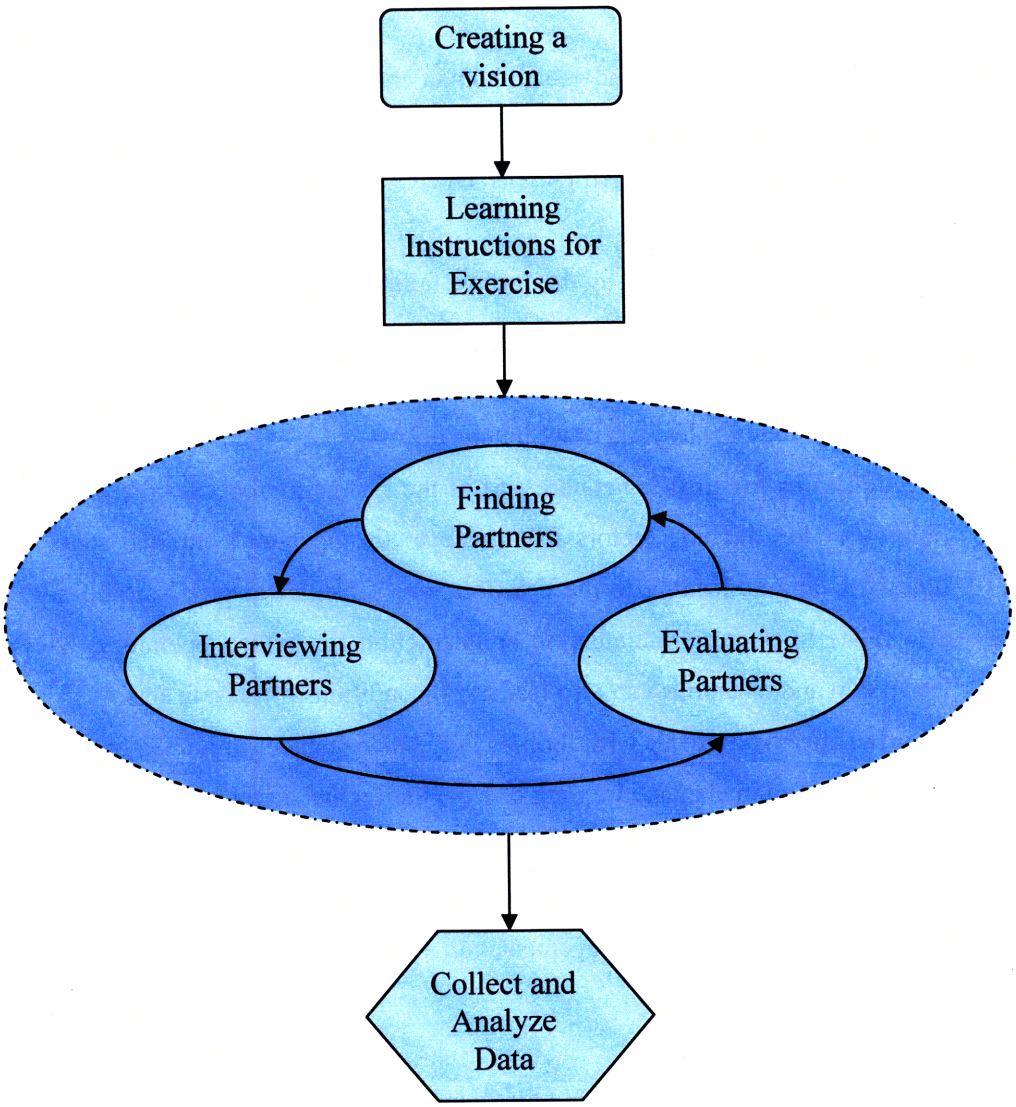


Figure 2-3: Overall design of teaching process in Leadership Module

2.3 Human Fallibility

It's an undeniable fact of life that humans make mistakes. Sometimes mistakes can be foreseen; sometimes they can be unpredictable. There are many examples of large-scale engineering designs that neglect to account for human fallibility [12]. Good system design should address the risks and minimize the impact of human errors. It was crucial that the technology support for the Leadership Module be robust.

Many models have been developed to understand human error. The Human Factors Analysis and Classification System (HFACS) was developed in military aviation training as a framework for identifying different causes of human error. The HFACS categorizes failures into four levels [14]:

1. **Unsafe acts** are manifested by the "active failures" of a system. They are connected with people who directly interact with the system, such as an aircraft pilot or air traffic controller, and stem from incomplete training, memory lapses, negligence, willful violation of rules, and other human errors.
2. **Preconditions for unsafe acts** are part of the systemic properties that lead to active failures. They include factors that relate to the operator's internal state, such as illness or fatigue, leading to the deterioration of mental or physical abilities. They also relate to poor communication or instruction for human operators. Ineffectual leadership and discipline can also lead to preconditions for unsafe acts.
3. **Unsafe supervision** deals with failures in the management of human operators. This may come from inadequate training or from ineffectual direct supervision.
4. **Organizational influences** can impact all unsafe acts in the system. All aspects organization architecture, from culture, ethics, and policies to procedures, regulations, and resource management, can affect human operators and lead to unsafe acts.

Understanding the HCAFS leads to the insight that human error can be caused by every level of a system hierarchy. Just because a mistake was caused by a person doesn't mean that all the fault and blame lies with that person. Robustness requires the architect and manager to address the systemic factors that lead to failures. Preconditions, supervision, and organization influences all belong to the "latent conditions" of failure in systems. One model of addressing human error in systems is the Swiss cheese model by James Reason [15], shown in Figure 2-4 To minimize human error, systems have multiple defensive layers, from training and supervision to rules and alarms. No defense is perfectly reliable, so each layer is analogous to a slice of Swiss cheese, with holes representing potential errors. The entire system defense, even with overlapping layers, can still have openings that allow human errors to have grave consequences.

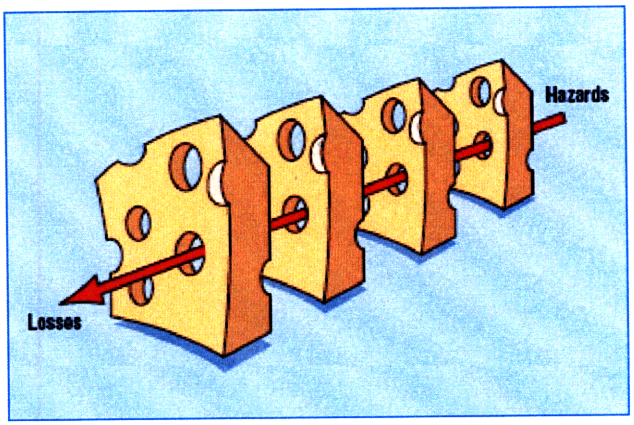


Figure 2-4: Diagram of Swiss Cheese Model (Reason 2000)

However imperfect a block of Swiss cheese may be, from an engineering perspective, a robust system is one that can tolerate all errors, either by preventing them or by fixing them after they occur. Technology plays a key role in system defense. When used wisely, technology can simply tasks, reducing the possibility for error. In addition, proper software can reliably identify and address errors. An effective defense involves heuristics that deal with predictable errors and adaptable, flexible human operators that solve unforeseen problems.

Chapter 3

The Dance-Card Generator

The goal of the Dance-Card Generator, or Dancer for short, is to randomly assign the entire class into pairs of leaders and followers for the entire game. The algorithm behind the Dancer generates a dance-card that tells every student his or her partner in every round of the activity. The input consisted of the IDs of the every student in the class, along with the total number of rounds. Section 3.1 describes the constraints and requirements of the Dancer. Section 3.2 gives the algorithm used by the Dancer, and Section 3.3 evaluates the output and user interface of the Dancer.

3.1 Algorithm Constraints

Every student requires a random partner for each round. Randomization eliminates the familiarity bias, where students try to pick their friends as partners. It also provides a well-connected network of students. Figure 3-1 demonstrates the differences between familiarity and randomness in two separate networks. Each network contains ten individuals drawn from two social groups, with each individual having three partners. Network A on the left features individuals pairing within their social groups—the resulting graph is almost disconnected between the two groups. Information between the two social groups in Network A must pass

between a single pair. Network B shows the result of random pairings, allowing information to be spread amongst the nodes of the graph.

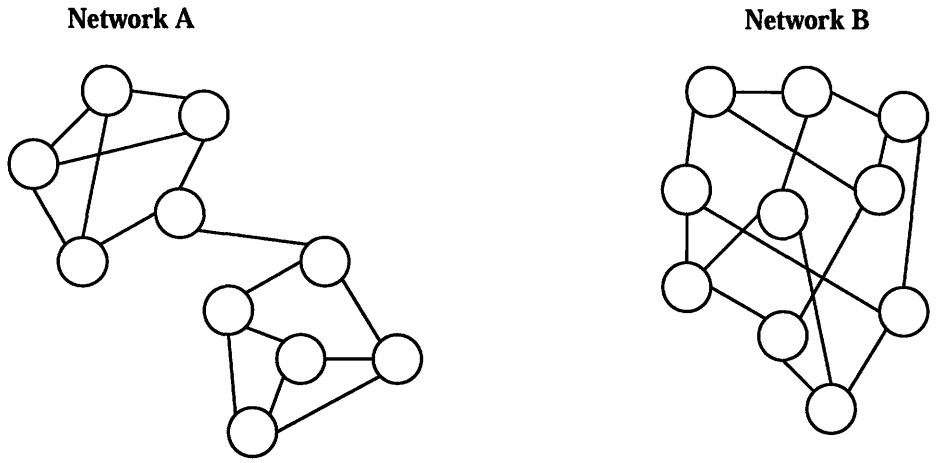


Figure 3-1: Comparison of two social networks. Each node represents a person and each arc represents an interaction. Each network has ten people and each person has three interactions.

The other main difficulty is that each student was not allowed to meet the same partner twice in the game. Therefore, over a dozen rounds, each person requires twelve unique assignments. In addition, since the number of students who actually participate in UPOP varies on any given day, we need to be able to quickly generate random matchings for any number of students. To this effect, our algorithm relies on the large number of students and small number of assignments—we expected a class to have about 120 students, participating in six rounds for both sides (as a leader and as a follower)—and we randomly matched students for each round, while continuously checking previous assignments and re-matching students who had conflicted pairings.

3.2 The Random Pair Algorithm

The algorithm is outlined here in pseudocode:

- Start with a 1-D array of leader IDs as input, and a blank 2-D array, with the follower and his partner for each round as output.
- For each round, arrange the input IDs in numeric order.

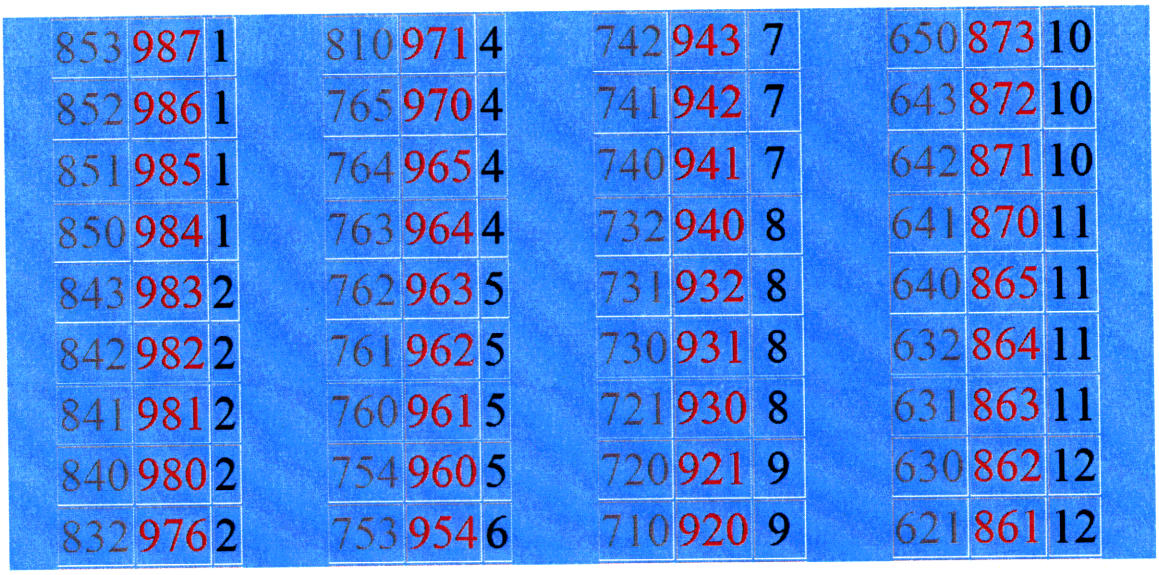
- For each ID in the output, randomly pick one of the remaining input IDs as the partner
    - Check to see if the input ID is present in a previous round
        - If no, then assign input ID to output as partner, and increment the indices of both arrays
        - If yes, then randomly pick another input ID to check
    - If after five tries, there is still a conflict with previous partners, the input array is reset and the process starts over for that round.
- With each successful assignment, the follower ID is added to a 2-D array of leaders with their assigned partners.

Through tests and actual runs, there were almost no instances of the Dancer having to reset the array due to too many conflicts.

3.3 Output from Dance-Card Generator

After this process is completed, each student will have twelve unique partners, one for each round. The advantage of the Dancer is its simplicity; it only requires a list of ID numbers and the number of rounds, so it can be run at the beginning of the exercise to assign partners for students. The assignments are stored as an XML document, along with the other data in the module, and displayed for students on the screens of the classroom using an XSL Stylesheet like in Figure 3-2:

---- Phase 1 Round 0 ---- Follower ABC meets Leader DEF at Table T



853	987	1		810	971	4		742	943	7	
852	986	1		765	970	4		741	942	7	
851	985	1		764	965	4		740	941	7	
850	984	1		763	964	4		732	940	8	
843	983	2		762	963	5		731	932	8	
842	982	2		761	962	5		730	931	8	
841	981	2		760	961	5		721	930	8	
840	980	2		754	960	5		720	921	9	
832	976	2		753	954	6		710	920	9	

650	873	10									
643	872	10									
642	871	10									
641	870	11									
640	865	11									
632	864	11									
631	863	11									
630	862	12									
621	861	12									

Figure 3-2: Sample output from random pair generator.

The disadvantage is that the algorithm immediately calculates all pairings for all rounds. If a student leaves in the middle, or if a student arrives late, the partners need to be manually updated, usually by swapping people at random. The process lacks flexibility in creating dynamic pairings among students. Since the number of changes is quite small, however, approximately 2–4%, the extra students were paired by hand.

The table assignments are based on the tables of the students with red IDs (the red IDs being leaders in the sample output above). Half of the tables are made up of red IDs and half are gray IDs, conveniently leaving enough space for all students. When the students read the screens to find their partners, only half of the students need to move, thus lessening the confusion during rotations of partners. Further complications came from students with gray IDs not being able to find their assigned tables.

Chapter 4

The Data Collector

Chapter 4 and Chapter 5 describe the Data Collector and Rank Calculator parts of the system respectively. Although they define different concepts, the Data Collector and Rank Calculator run on the same software program. Hence, there will be overlap between these two chapters.

Figure 4-1 shows the process flow of the entire software system. The input from students comes in the form of scantron forms and is fed into an optical mark reader (OMR). The design of these two components will be explained in Section 4.1 and Section 4.2 respectively. The OMR is run by a software program that converts the form data into raw matrices stored as XML documents, which is described in Section 4.3. The other components will be covered in Chapter 5.

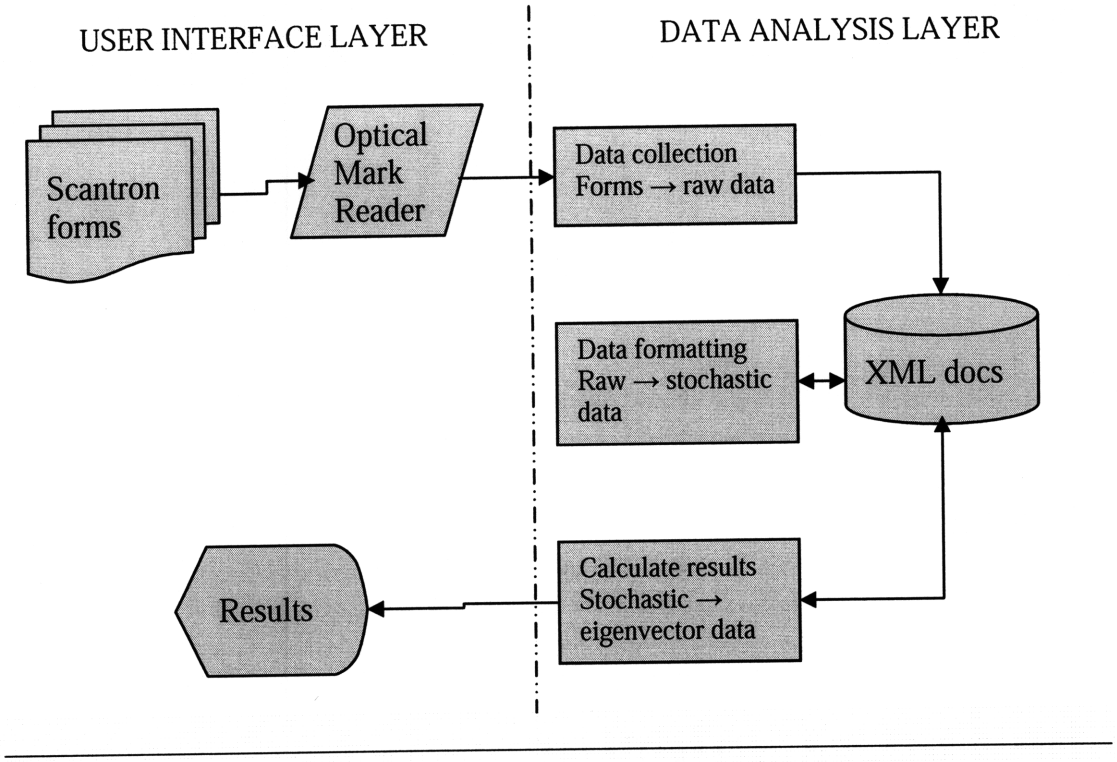


Figure 4-1: Software and hardware process flow for the UPOP Leadership Module

4.1 Optical Mark Reader

After discussing different options for input of student evaluations, we chose the optical mark reader (OMR) as the most cost-effective. Laptops were too cumbersome, and the handheld voting keypads in the room did not fulfill the needs of the module. Data input via OMR is a method that students were familiar with, having taken plenty of exams on scantron forms in the past. An added benefit of using scantron forms was the paper documentation of student evaluations. We settled on the BubbleScan, an OMR built by Academy Technologies[17], as the optimal solution. The alternative manufacturers, such as Pearson Education[18] and Scantron Corporation[19], featured expensive, industrial machines. The BubbleScan read forms manually, but it had no mechanical or moving parts, and can read up to thirty forms a minute, which was sufficient for our needs. Finally, because Academy Technologies is a relatively small company with few customers, they always had technical support available. Their technical support proved invaluable in customizing the data collection software.

4.2 Interview Game Forms

The forms were designed for the optimal combination of information, clarity, and reliability. An example form is given in Appendix A, with the original form as A1. The form read by the optical mark reader occupied half of a sheet of standard letter paper. It is printed on special paper purchased from Academy Technologies, which is perforated in the middle along the vertical axis. There is a line for the student's name and table, to easily identify students in case the form failed to be read. Each form contains approximately sixty rows (as shown in the original form), with six columns of bubbles per row. For simplicity, the entire exercise takes place on one scantron form. Therefore, the form is divided into sections, with each section containing information for one of the six rounds of the exercise.

Each student is given a unique ID to be used with the data input software. The structure of the ID was determined by space constraints of the form. The easiest identification—a combination of table number and student number—would have required three rows for the table number and two rows for the student number—five rows per round. The ID structure that occupied the least space consisted of only two rows, with ten digits labeled from 9 to 0. Each ID consists of three non-repeating, descending digits. This creates (10 choose 3) = 10!/[(10-3)! * 3!] = 120 available IDs. Since each UPOP class cannot exceed 117 students, this ID structure is sufficient.

On the form, each student bubbles in his own ID at the top, and then for each round of the game, his partner's ID and his evaluation of that partner. The evaluation is determined on a scale from 1 to 6, because we could have at most six bubbles per row. At first, students were only asked to score their partners between 1 and 6, 6 being the highest score. During a test run, it was discovered that students preferred to rate all of their partners highly, scoring a median between 4 and 5. In the eigenvalue analysis, the scores are normalized, so only the difference between scores would be measured. Therefore, to increase the range of data, students are asked to rank their partners after meeting all of them, giving a score of 1 to the best leader/follower, and 6 to the worst. Appendix A2 shows the form at this stage.

At first, the other side of paper was merely intended as a cover sheet, to afford the students privacy in their evaluations. The students found it helpful, however, to take notes on the blank side of the paper during the exercise. It was then decided to add blanks for students to write their partner's IDs and to take notes about their interviews. Finally, to minimize risk of errors,

students are told not to bubble in anything until the conclusion of the game, to minimize risk of bubbling mistakes. After testing pencil, pen, and ink marks on the scantron reader, it was discovered that only permanent marker could be reliably read by the OMR. Appendix A3 presents the final form used in the 2007 Leadership Module.

4.3 Data Collection Software

The software program was written in Java for two reasons: first, Java is a popular language with many resources available for customized programming. Second, we could foresee this program being used by other organizations interested in social networking studies, and they would more likely to be able to run a Java program than say, a MATLAB program.

The data analysis program consists of three phases, as shown in Figure 4.1—reading and rearranging the output from the OMR into Java objects, converting raw data into stochastic matrices, and then finding the eigenvectors corresponding to the dominant eigenvalues of the matrices. Although the BubbleScan has its own Survey Wizard software to interface with the OMR, we decided to read from the OMR directly in Java, giving us the ability to test the output for form entry errors, allowing greater flexibility and robustness. The output from the OMR is read via the serial port, which we accessed using the javax.comm package. After converting the matrices to stochastic form, we applied the power method to find the dominant eigenvectors, similar to Google's PageRank algorithm, which is described in Chapter 5.

At the end of the exercise, the forms are collected from students and fed into the optical mark reader. This process unavoidably requires man-power, from gathering forms and feeding them through the OMR. In addition, the operator of the OMR needed to be trained to minimize errors in the data. First, the forms needed to be cleanly separated along the perforated edge. The operator also checked for forms folded, torn, or rolled incorrectly, which would have gotten stuck in the machine.

It was discovered during test runs that some students invariably forgot to bubble in their IDs. Since the software only identified students by ID number, incorrectly-filled forms were rejected by the data collection software, and the operator had to manually input the data from those forms. Hence, all data read by the software was stored in XML documents. XML was flexible, as well

as being easy to read and modify. While forms were being processed, the problematic forms were placed on the side, and added to the XML document.

The software for the optical mark reader, called Survey Wizard, was originally written by Academy Technologies. The Survey Wizard could read the forms and produce standard output (in the form of choices 1-6 for each line) on a text file or comma-separated file. The Leadership Module has different requirements from regular test or survey applications, however, and requires customized software. The goal is to have software that checked in real time for human errors, using heuristics to correct for minor errors and eliminating forms with major errors.

The software checks for the following errors in real time:

- If any score was greater than 6, then that round was invalidated.
- If any ID was greater than 987 or less than 321, then that round was invalidated.
- If the team color didn't match the Leader/Follower status—in other words, if an individual used the incorrect form for his role—then the form was invalidated.

Chapter 5

The Rank Calculator

The third component of the software program calculates rankings of students based on their peer evaluations. Section 5.1 introduces social network theory as a foundation for data analysis. Section 5.2 gives a summary of the PageRank Algorithm, which was a very interesting application of eigenvalue analysis to a networking problem. Section 5.3 explains the algorithm and software in detail, describing the data formatting and power iteration method for ranking students, and concluding with a small example to illustration the Rank Calculator.

5.1 Social Network Theory

To effectively analyze the leader-follower data and draw conclusions, one must be knowledgeable in the field of social network theory. In the past few decades, social network analysis has evolved from a little-known discipline to one of the most popular topics in the social and behavior sciences. The fundamental difference of social network analysis from traditional social science approaches, and hence its appeal to researchers, is its focus on relationships between people, instead of people as independent objects. Consequently, social network analysis has its own terminology, with a problem being defined partly in terms of actors, relations/ties, and groups/networks. In order to scientifically study the results from the Leadership Module, we model the data as a social networking problem [11].

5.2 The Google PageRank Algorithm

The PageRank algorithm was developed by Larry Page and Sergey Brin while researching at Stanford University in 1998. It models the World Wide Web as a directed graph, with websites as notes and hyperlinks as edges. Each link from site A to site B is essentially a recommendation of B by A. The strength of the recommendation depends on A's authority, which stems from the links leading to A by other sites. A recommendation from a well-known site is worth more than one from an unpopular site. The strength of the recommendation also depends on the total number of recommendations given by A. If A is more generous with its links, then each link is worth less. In short, a website is important if it's linked by other important websites [16].

The Web can also be modeled as a Markov chain, where a PageRank of a website P is defined as the sum of PageRanks of all websites pointing to P, divided by the total number of links of websites pointing to P. It can also be described by the random surfer model, where an individual surfs the Web by randomly clicking on links to new website. In a steady state, the probability of landing on a particular website is equal to the PageRank of that site. The PageRanks are calculated using the power iteration method in eigenvalue analysis, and the PageRank vector is the eigenvector associated with the dominant eigenvalue of a dense, stochastic, primitive matrix called the Google Matrix. The Google Matrix is structured to optimize the convergence – according to Google, 50 iterations of the power method are enough to ensure convergence [16].

5.3 Data Analysis of the UPOP Leadership Module

The mathematics in this project is based on the following assumption: good leaders are better at identifying good followers than poor leaders. Likewise, good followers are better at identifying good leaders than poor followers. During the activity, students evaluate each other how their skills at leading and following. There are multiple rounds, so each student is evaluated multiple times by his or her peers. These scores are the basis on which we find the best leader and follower.

5.3.1 Stochastic formatting

Before the eigenvalue analysis, the data must be converted to stochastic form. After the raw data is compiled into an XML document, it is stored two sparse matrices. The first matrix L' features leaders evaluating followers; if given n leaders and m followers, then L has n rows and m columns: each element contains a score from 1-6 if the leader interviewed and then scored the follower, otherwise it contains 0. The second matrix F' contains the same data, only for followers evaluating leaders. In stochastic form, the sparse matrices are converted to dense matrices, with the sum of each row being equal to 1. The matrices are converted using the following algorithm:

- For L', for each row r, calculate the sum of the scores and denote t[r] as the sum of scores for row r
- Denote p[r] as the total number of partners (i.e. nonzero elements of r)
  - For each element i of r, if i > 0, then
    - i = i / t[r] * p[r] / n
  - Otherwise, if i = 0, then i = 1 / n
- Do the same for F'
- Denote the resulting matrices L and F

Besides using the raw scores 1-6 in the stochastic matrix, the stochastic data could have been modeled in other ways. An interesting question was if instead of 1-6, the data consisted of scores of 0-5. Since the total score is smaller, there is a higher differential between consecutive scores. In other words, there's a greater marginal loss from 0 to 1 in the new scale than from 1 to 2 on the old scale. After performing the eigenvalue analysis, there was almost no difference in rankings among students between the zero-based linear algorithm and the original linear algorithm, as shown on the left side of Figure 5-1. The Spearman's rank correlation coefficient was 0.998.

Another algorithm would map the raw data to exponential weights. Instead of 1-6, students would be allocated scores of 1, 2, 4, 8, 16, 32. In other words, being ranked first would be worth twice as much as being ranked second. Calculating the eigenvalue rankings and plotting the exponential algorithm with the original linear algorithm on the right side of Figure 5-1, there is a very high correlation between the two methods, with a Spearman's rank correlation coefficient of 0.753. Hence the eigenvalue analysis is robust to changes in data, as long as the relative rankings are preserved.

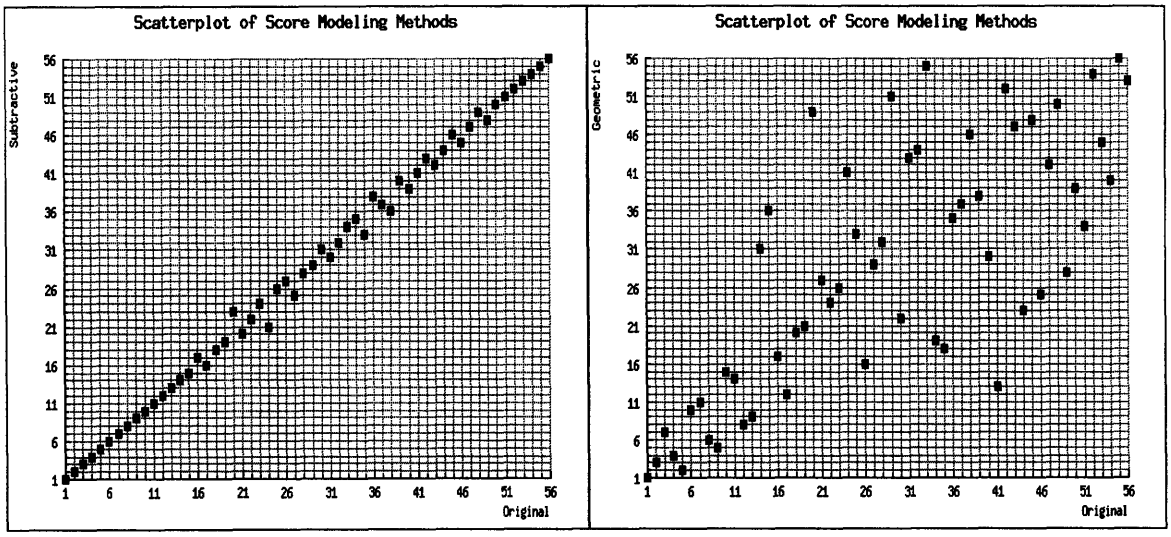


Figure 5-1: Scatterplots of rank correlations, for the zero-based linear and exponential methods compared to the original linear.

5.3.2 Eigenvector Analysis

After converting the data to stochastic form, we have two matrices—one of leaders evaluated by followers, which will be called *L*—plus one of followers evaluated by leaders, called *F*. We then apply the power method to these two matrices to find the dominant eigenvector. After multiplying *LF* by itself *k* times and then multiplying by a vector *b* (composed initially of equal, normalized values), we have a sequence that has converged to the dominant eigenvector. The same method is applied to *FL*. For this method to work, we follow the assurance inherent in the power method—the eigenvalue 1 is greater in magnitude than all the others.

The algorithm is as follows:

```
// output: l and f eigenvectors, corresponding to b, described above
// input: L and F are sparse 2-D matrices, as described above
int k = 0;
double residual = 1;
double epsilon = .00001;
for #followers
      initialize all elements of f to 1;

while (residual >= epsilon || k > 100) {
      store f as f_previous;
```

```
        k = k + 1;
        f = f * F;
        f = f * L;
        f = f * (1 / #followers);
        residual = norm(f - f_previous);
}
l = f * F;
l = l * (1 / #leaders);
```

This algorithm is similar to the PageRank algorithm used by Google to compile its search results, as well as the HITS algorithm developed by J. Kleinberg [16]. Each leader is evaluated by a number of followers, and the leader also evaluates the followers. The best leaders and followers are those with the highest scores. Both the PageRank and HITS algorithms use complicated weighing systems to give certain websites more authority than others. We decided to start with all of the students weighted evenly, and allow the scores to converge on a steady state.

Another point to consider is the evaluation system used by students. The most simple one, and one that was used in the first trial simulation, is a binary ranking system—if a leader is considered above average, he/she will receives a 1, otherwise a 0. A more favored approach involves ranking on a scale of 1–6. A student can use each score only once to ensure that there are no ties in evaluations. This ranking method gives us much more information, but also greater variance in individual interpretation of rankings. Future considerations include evaluations with multiple dimensions, based on the different characteristics of good leaders.

5.3.3 An Example of Eigenvalue Analysis

Figure 5-2 gives an example of a game with 6 students and 3 rounds. The leaders have red nodes and IDs 1, 2, and 3. The followers have gray nodes and IDs 4, 5, and 6. Each student has partnered with the 3 students of the opposite role, represented by arcs between nodes. Their rankings for each of their partners are next to their nodes and by each arc. Figure 5-3 shows the same information produced by the Data Collector in an XML document. Note that the scores from the Data Collector have been inverted, so that a rank of 1 translates to a score of 3 and vice versa. Figure 5-4 presents the final scores generated by the Rank Calculator. From the eigenvector scores, the leaders would be ranked 2, 1, 3, and the followers would be ranked 4, 5, 6.

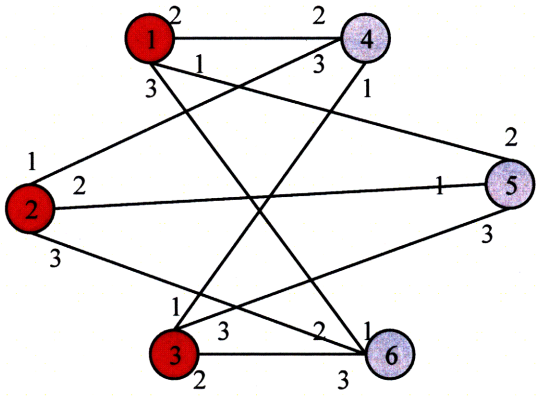


Figure 5-2: Example of interview game with 3 leaders in red, 3 followers in gray, and 3 rounds. Each student is a node with an ID 1-6, and each arc represents a ranking from that student to his partner.

```
<Leader ID="1">
  <Follower ID="4" score="2"/>
  <Follower ID="5" score="3"/>
  <Follower ID="6" score="1"/>
</Leader>
<Leader ID="2">
  <Follower ID="4" score="3"/>
  <Follower ID="5" score="2"/>
  <Follower ID="6" score="1"/>
</Leader>
<Leader ID="3">
  <Follower ID="4" score="3"/>
  <Follower ID="5" score="1"/>
  <Follower ID="6" score="2"/>
</Leader>
```

```
<Follower ID="4">
  <Leader ID="1" score="2"/>
  <Leader ID="2" score="3"/>
  <Leader ID="3" score="1"/>
</Follower>
<Follower ID="5">
  <Leader ID="1" score="2"/>
  <Leader ID="2" score="3"/>
  <Leader ID="3" score="1"/>
</Follower>
<Follower ID="6">
  <Leader ID="1" score="3"/>
  <Leader ID="2" score="2"/>
  <Leader ID="3" score="1"/>
</Follower>
```

Figure 5-3: Example data in XML format.

```
<leaders>
  <Leader ID="1" EigScore="0.36574074074074076"/>
  <Leader ID="2" EigScore="0.4675925925925927"/>
  <Leader ID="3" EigScore="0.16666666666666669"/>
</leaders>
<followers>
  <Follower ID="4" EigScore="0.43904320987654316"/>
  <Follower ID="5" EigScore="0.36651234567901236"/>
  <Follower ID="6" EigScore="0.19444444444444445"/>
</followers>
```

Figure 5-4: Example results generated from Rank Calculator. "EigScore" represents the eigenvector scores. IDs with higher scores have more importance in the network.

Chapter 6

Implementing the First Leadership Module—January 2007

The first iteration of the leadership module focused on gathering the richest data set. To achieve this, students each took part in 12 rounds of visioning and collaborating, with evaluations after each round. Each round consisted of a three minute interview between a leader and a follower. After the interview, the students evaluated each other on a scale from 1 to 6. The overall goal of the exercise was to use the Rank Calculator to rank the students and find the ones rated highest by peers. The Dancer matched students each round.

The first two sections of this chapter introduce transformational leadership theory and leader-member exchange theory as background for teaching leadership to students. The remaining sections describe the execution of the UPOP Leadership Module in 2007. Section 6.3 explains the hacks that motivated students. Section 6.4 concludes the software design from previous chapters with a user interface that manages the entire process. Section 6.5 describes and evaluates the exercise, including student feedback and recommendations.

6.1 Transformational Leadership Theory

One of the central concepts of the Leadership Module is the theory of transformational leadership. The two most well-known proponents of this theory are James MacGregor Burns and Bernand Bass, each with his own version. Burns, in 1978, first proposed that leaders inspire followers by appealing to higher ideals and moral values. This approach is effective because the right social and spiritual values are hard to deny, and are applicable to large groups of people. Moreover, appealing to social values encourages teamwork and collaboration, which is much more useful than appealing to selfish or personal motives (9). Bass, who worked with Burns in studying leadership, countered that leaders should place more importance on raising their follower's awareness of tasks and importance of performing well. They also stress intellectual motivation and idealized influence. In short, Bass used similar methods as Burns but without the moral considerations; for example, he attributed transformational leadership skills to Adolf Hitler (10).

In the Leadership Module, then, the goal of a leader is to inspire followers to participate in the leader's activities. To achieve this requires identifying with the follower's needs and values, and may require appealing to higher ideals. Leaders also need to have charisma and good communications.

6.2 The Relationship between Leaders and Followers

First proposed by Graen and Cashman in 1975, the Leader-member exchange (LMX) theory separates the leader-follower relationship into two categories: high-exchange and low-exchange relationships. In high-exchange relationships, both the leader and follower have incentives to work harder than required for completing tasks. The leader benefits from having a loyal, committed follower, and the follower benefits from having the leader's positive influence. Conversely, in low-exchange relationships, usually confined to menial jobs, leaders and followers have no incentive to go beyond formal role requirements.

It has been shown that the follower's beliefs and assumptions about effective leadership, known as implicit leadership theories, heavily influence the follower's evaluations of the leader [7]. Through personal and relational experiences, individuals develop opinions and stereotypes of

leaders in specific positions, such as at different levels of an organization or across different industries.

Many of the interactions between leaders and followers involve each party trying to influence how they are perceived by the other, known as impression management [8]. Wayne and Ferris conducted a study in 1990 in which followers tried to make the best impression in three areas: exemplification, ingratiation, and self-promotion. Exemplification focuses on amplifying one's dedication and commitment to one's job. Ingratiation, not surprisingly, involves building productive relationships with one's superiors. Self-promotion focuses on showcasing one's achievements, skills, and competence.

6.3 Creating a "Hack"

To motivate the exercise, students created and presented hacks. The MIT culture defines a hack as "a clever, benign, and ethical prank or practical joke, which is both challenging for the perpetrators and amusing to the MIT community." One well-known example is an MIT police car being placed on the great dome, as seen in Figure 6-1.

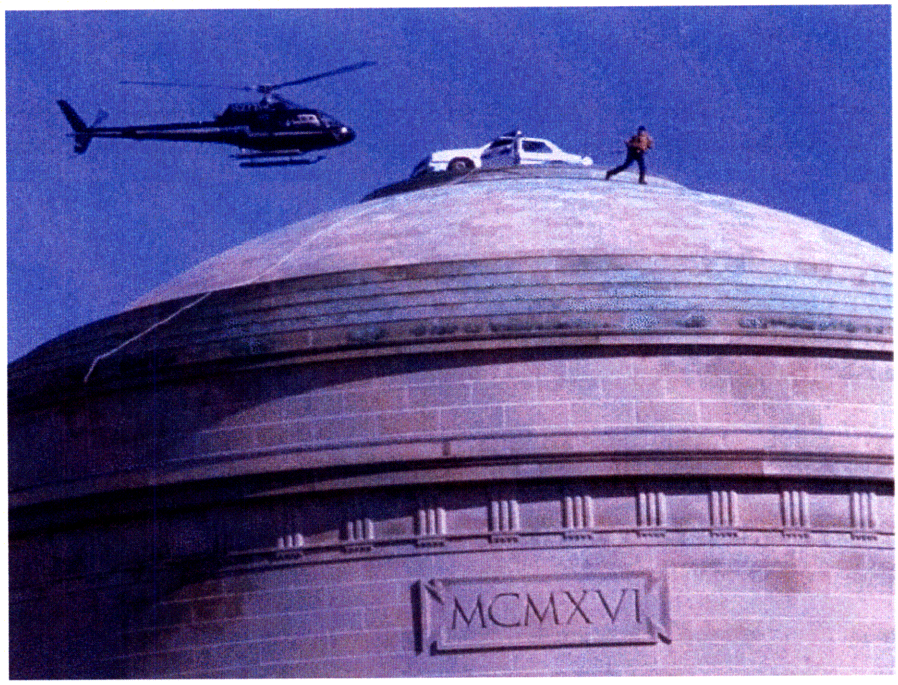


Figure 6-1: Example of MIT hack – police car mysteriously placed on great dome.

6.4 Process Workflow

Chapters 3, 4, and 5 together describe the software tools used in the 2007 Leadership Module. In the final system, these tools are integrated with running the interview game and reading forms. Currently a simple selection menu, seen in Figure 6-2, allows the user to easily select which software function to use. When a form is read, the data is tested for every possible error, with faulty data automatically removed from the input stream. The result is a streamlined process that efficient collects data and isolates erroneous forms. Future work can improve the user interface, perhaps creating a GUI to allow users with little experience to easily operate the program. An improved user interface would also enhance presentation of results, permitting better integration with the rest of the leadership module.

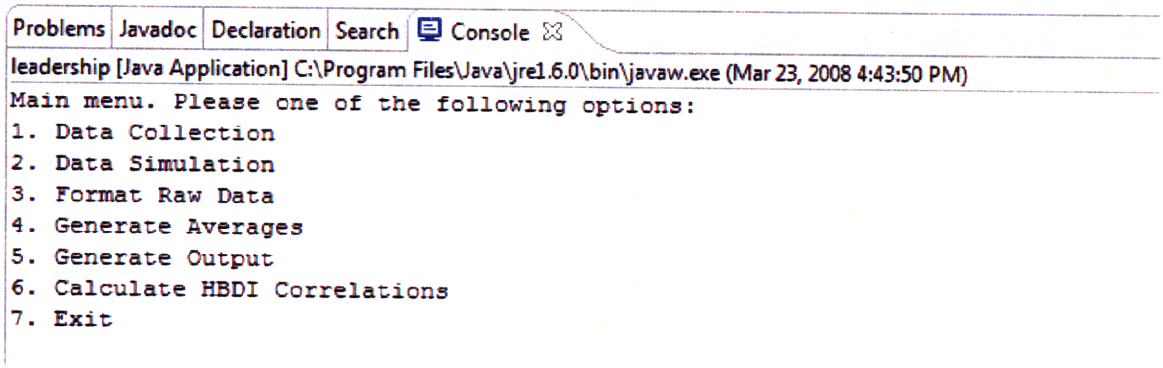


Figure 6-2: Software user interface with Optical Mark Reader and analysis tools.

6.5 Procedure and Results

The Leadership Module was run on January 23rd and 30th, 2007. Despite the planning and preparation of the exercise, there were many unpredictable variables that arose in classes of 120 students. This section performs a step-by-step analysis of the performance of the random pair generator, forms, optical mark reader, and data collection software.

Before the students arrived, IDs were written in red or gray numbers on sticky nametags. Since only red IDs would be matched with gray IDs and vice versa, the IDs were distributed at the table in alternating colors. Only after the students arrived and the leadership module started, were the random pairings generated. The goal was to have the most accurate count of students possible. Some students still arrived late and had to be paired manually.

During the instructions phase, the students participated in a warm-up interview round, where they partnered with student at their original table and practiced pitching their hacks to each other. After the practice interviews, the students spend a few minutes giving each other feedback. The students all welcomed the opportunity to improve their presentation skills. Also during the instructions phase, the students were given detailed directions for filling out forms, because accuracy was essential and students couldn't fix errors made in permanent marker. If at any point the students made a mistake on their form, they were told to transfer the correct information to a new form and discard their old form.

After the instructions were given, the students were told to find their randomly assigned partners. At this point, approximately 120 students had one minute to find their partners, with half of them wandering around the room. The first problem came when students couldn't find their assigned tables because not all of the tables are positioned in order. To fix this first order of confusion, the mentors at each table were asked to stand up and hold their table-signs above their heads. Next, there were the one or two students who came in late, and whose IDs weren't included in the random pair generator. This presented difficulties, since the late students had be manually paired each round, but they were reasonably few. The final problem came when there happened to be an odd number of total students. In this instance, one of the UPOP alumni helpers, known as runners or production assistants, filled in the role of a leader or follower.

The confusion died down after the first or second rounds, and the students settled into the pattern of interviewing and role-playing, evaluating their partners by taking notes, then moving on to new partners. For the first six rounds, students only played one role—leader or follower. After six rounds, they ranked their partners in terms of effective leadership/followership, turned in their forms, and switched roles for another six rounds. In total, students each participated in twelve actual interviews of three minutes each: six as leader, six as follower.

After the first six rounds, the forms were collected by assistants, and then fed through the OMR. At this time, multiple problems were discovered in the data collection process. Although designed for robust error-handling, the data collection software was actually quite fragile, and could abort with any unexpected signals from the OMR. For instance, there were forms that had been incorrectly perforated and unable to fit in the OMR. When they became stuck, the data collection software timed out and aborted its current process. All of the data previously read but

unsaved was lost. This was fixed through a time-consuming process, where forms were read in groups of 20, and each group's data was saved in a separate XML document. If one group had a faulty form, that form was removed and the other forms in the group were rescanned. The data from the faulty forms would need to be entered by hand.

The leadership module was well-received by students. They particularly enjoyed the activity and opportunity to improve their communication skills. Many of them had fun specifically coming up with hacks and persuading their classmates to agree to work with them. Students complained, however, that there wasn't enough personal feedback for everyone. It was decided not to tell every student his or her ranking, for fear of discouraging students ranked at the bottom. Only the highest-ranked leaders and followers were singled out and questioned about their methods and goals. The majority of students found the "winners' panel" to be unhelpful, however. The feedback prompted us to remove the winners' panel from future Leadership Modules.

Chapter 7

Implementing the Second Leadership Module—January 2008

The second leadership module used many of the components in the first iteration, but it also tried to improve and refine components. Recognizing that the primary aim of the module was to teach students about leadership, the new iteration eliminated the random pairing and ranking algorithms to streamline and simplify the process.

One of the new tools explored in the 2008 Leadership Module was the Hermann Brain Dominance Instrument, which is explained in Section 7.1 and will be applied to the new form design in Section 7.5. Sections 7.2–7.8 describe changes made to the implementation of the 2007 Leadership Module. The final results and evaluations from the Leadership Module are presented in Section 7.9.

7.1 The Hermann Brain Dominance Instrument and Whole Brain Model

The Hermann Brain Dominance Instrument (HBDI) is used by UPOP to help students learn about their thinking styles. Developed by Ned Hermann in 1979, the HBDI uses the theory of the brain composed of four separate structures—the two hemispheres and the two limbic structures, demonstrated in Figure 7-1. Through a series of cognitive analysis questions, the HBDI attempts to measure the strength of an individual's preference in the four quadrants [13]:

A: Logical, analytical, fact-based, quantitative

B: Organized, sequential, planned, detailed

C: Interpersonal, feeling-based, kinesthetic, emotional

D: Holistic, intuitive, integrating, synthesizing

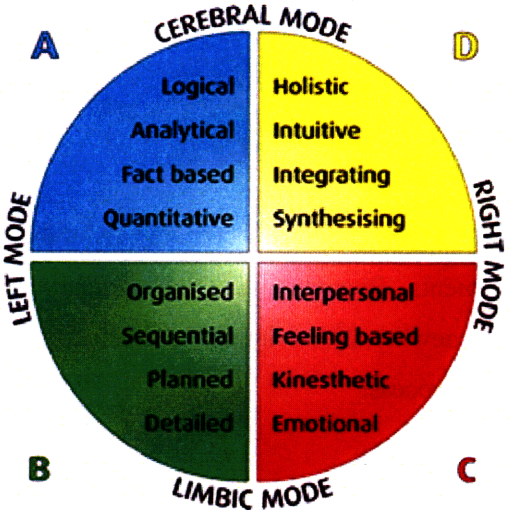


Figure 7-1: Whole Brain Model for HBDI (Hermann 2007)

7.2 Creating a Vision

The first component, as shown in Figure 2-2 and Figure 2-3, asked students to think of a compelling vision that they would present to peers. This was accomplished through short class sessions and a "Visioning Worksheet" that presented examples and qualities of good visions, as well as asking students about their personal values and goals. In the previous leadership module, students only used hacks in their interviews. Instead of limiting the visions to hacks, this year the

students could present any topic that was of interest to them. The rationale was that students would communicate more effectively when they were passionate about their message, and there would be a richer set of discussions with the inclusion of visions with social impact.

7.3 Identifying Students and Partners

During the exercise, students were given IDs in the form of three non-repeating, descending digits, just like the IDs in the 2007 Leadership Module. The students were also instructed on the roles of visionary and collaborator. Each student found a partner, and each partner in the dyad played a different role. The visionary explained his or her vision and tried to persuade the collaborator to join his or her team to implement the vision. The collaborator worked with the visionary to improve the vision, providing feedback and new ideas, while persuading the visionary that he or she would be the best person for the visionary's team. The names "visionary" and "collaborator" were chosen over "leader" and "follower" because the latter had hierarchical connotations. Using visionary and collaborator most effectively captured the idea of a partnership between equals, analogous to the founder of a startup and the venture capitalist providing funding and support.

7.4 Student Movement between Rounds

Each student participated in eight rounds, with each round taking four minutes to complete. After experimenting with different round lengths and quantities, this choice presented the best balance between collecting sufficient data and not overwhelming the students. During the rounds, each student alternated between the role of visionary or collaborator. Alternating allowed students to observe and learn from the behavior of other visionaries, allowing them to iteratively improve their own communication styles.

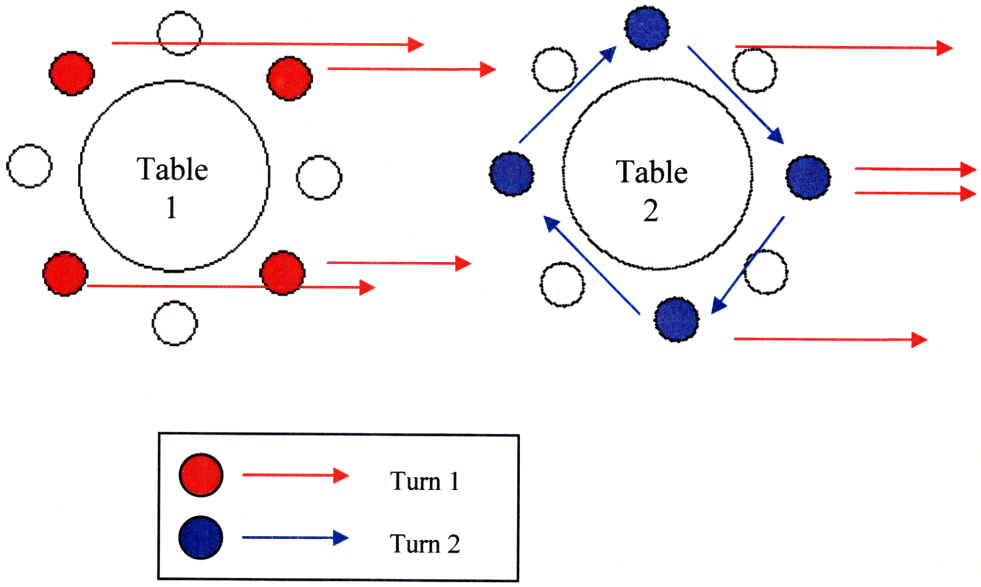


Figure 7-2: Movement between rounds as students find new partners

Figure 7-2 shows the movement of students between rounds. Each round, only half of the students moved from their seats to find their partners. The movers were directed towards the neighboring table. Since all of the tables are numbered, the students at Table i moved to Table i + 1. Then, the students found their own partners at the new table, with the only constraint being that no student may have the same partner twice. With approximately 120 students and each student having eight partners, this constraint wasn't difficult to satisfy. The majority of the students found their partners quickly and effectively. This algorithm provides the benefits of simplicity and efficiency, at the cost of precise pairings.

7.5 Peer Evaluations

The students each carried a paper form, used to evaluate his or her partners (see Appendix B2). The evaluation form is the interface between student input and the data analysis software. Thus, the form was designed for usability and learnability. The design is similar to the forms used in 2007. Each question on the form has its own instructions. The students fill in their three-digit

IDs at the top, as well as the IDs of their partners each round. Then, they evaluate their partners with two questions: First, how effective was their partner's communication style? Second, how effective would they and their partner be, working in a team? The two questions endeavored to focus the students on communication and teamwork, essential components of leadership.

In the first week of the 2008 Leadership Module, students were also asked to map their partner's leadership and followership abilities to the four HBDI quadrants, as seen in Appendix B1. The goal was to In the second week, the HBDI questions were discarded in favor of simplicity and ease of use: each form could be used for four rounds, allowing each student to only use two forms for the entire exercise.

7.6 Collecting Data from Evaluation Forms

After the students complete their forms, they handed them to their mentors and assistants, who fed the forms into an optical mark reader (OMR) for data collection. This part used the same hardware—Bubblescan—from the 2007 Leadership Module, as discussed in Section 4.1.

7.7 Software Data Analysis

Data analysis consisted of three parts. The first subcomponent read the data directly from the forms to store as raw data. The second subcomponent translated the raw data into a separate format, allowing ease of manipulation. The final subcomponent calculated the averages of the student scores and presented the results for each student. Through the entire process, data was stored in XML documents. Each subcomponent read its input from an existing XML, and outputted its results in a new XML file. This method was superior to storing the data within the program memory, because it was sometimes necessary to modify the data by hand, in case any errors in data collection were missed by error-checking. It was also easier to use than managing a database system, and fulfilled the needs of the activity in the simplest manner.

7.8 Feedback for Students

The 2008 leadership module had a different goal than the 2007 leadership module. Instead of using evaluations to rank visionaries and collaborators, the software calculated the averages of each student's scores. After the forms were read by the optical mark reader, the student evaluations were rearranged so that each student entry included the evaluations *of* that student, instead of evaluations *by* that student. Afterwards, the software merely calculated the averages for each student, differentiating between visionary and collaborator.

After calculating average scores for each student, the software matched each student's ID with his or her name in a text document and then printed the name and scores for each student. The printouts were given to students as feedback for communication and teamwork.

7.9 Results and Recommendations for Future Work

In January 2008, the leadership module was run twice, once for each session of UPOP. The module began with some leadership theory presented by professors from the Sloan School of Management, followed by a tutorial of the exercise and the exercise itself. In each session, students participated in eight rounds, each round taking four or five minutes to complete. After each round, they evaluated their partner on communication and teamwork abilities. The students finished the module with a discussion about the components of successful leadership and received their feedback at the end.

The primary purpose of the leadership module is to teach students about leadership and the importance of visions. Therefore, at the conclusion, students were asked to provide feedback about the module, including what they learned and whether the module fulfilled its purpose. The majority of students stated that they learned about the importance of communication and passionately selling one's vision. Out of 179 UPOP evaluations on the day of the activity, 60 said that the leadership module was the most valuable experience that day. 35 said the leadership module was the least valuable experience, or actively disliked the experience. The remainder did not provide positive or negative feedback about the leadership module.

With regards to negative feedback, any students complained that the process was unnecessarily complicated, and the instructions weren't sufficient. Students also found the feedback lacking. Some expected more helpful or meaningful advice, such as the feedback given by the HBDI

results. Because the evaluations between students were subjective, there was no structured interpretation of the evaluation scores. A possible solution to this would be to have each student carry an envelope, and for the partners to insert personal feedback in the envelope, justifying their evaluations.

Chapter 9

Conclusion

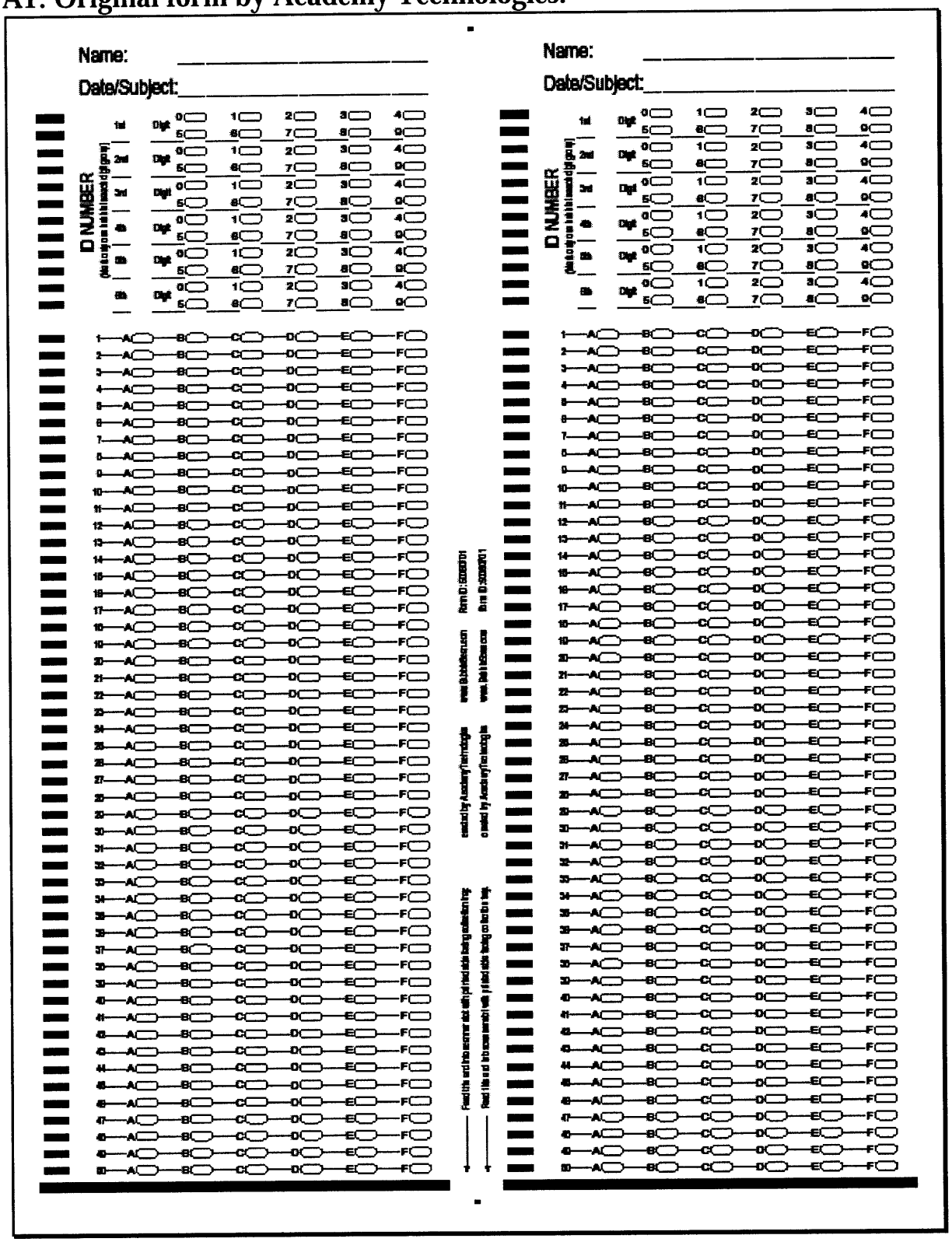
Unlike many of its peers in the Department of Electrical Engineering and Computer Science at MIT, this thesis is not primarily solving a technical problem. Instead, the challenges faced in the design of a software system for interviewing large groups mainly deal with people. It follows the mission of its main beneficiary, UPOP, closely—bridging the gap between academia and industry by applying theories to real world challenges. There were no new theoretical concepts discovered or proved in this thesis. Instead, its contribution comes from using existing technology and software in a new application.

Another interesting point is the iterative development of the UPOP Leadership Module. A great deal of time and effort was invested into the 2007 Leadership Module. The Dance-Card Generator and Rank Calculator were novel ideas, but they were discarded in 2008 because they did not fit the learning objective of the Leadership Module. The Rank Calculator created winners and losers among the students, but leadership isn't about winning or losing. Feedback was the most important learning factor for students, and since we didn't want to give negative feedback to the lowly-ranked students, we decided not to use eigenvalue analysis. Without the Rank Calculator, it wasn't important to have a well-connected graph. Therefore, students could find partners using a simpler, more flexible process. In general, the Leadership Module strove to become easier to learn for students, while retaining its visioning and relating activities.

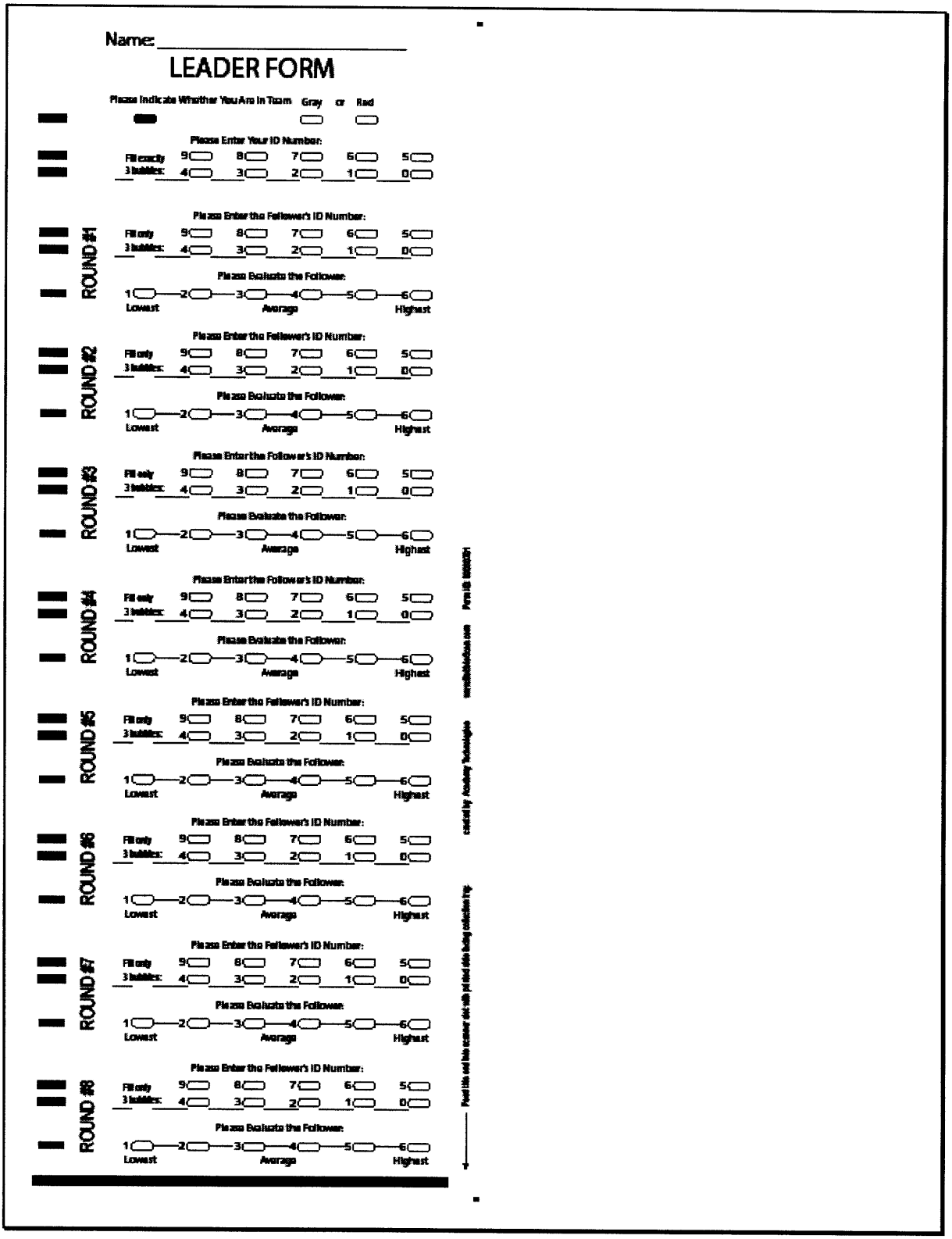
There are many other applications for this system. The speed dating industry would benefit from utilizing technology, in the form of scantron forms and readers, to streamline the matchmaking process. The same technology could make job interviews faster and more efficient. With a scantron form for evaluation and a software algorithm to determine results, job candidates could be notified minutes after completing their interviews. There are also many academic situations, such as classes and experiments, which could benefit from the quick and efficient feedback achieved with scantron forms.

Appendix A: Forms for 2007 Leadership Module

A1: Original form by Academy Technologies.



A2: Modified form with student ID and evaluations.



Name: _____

LEADER FORM

Please Indicate Whether You Are In Team Gray or Red

Please Enter Your ID Number:

Fill exactly 3 bubbles: 9 8 7 6 5 4 3 2 1 0

ROUND #1

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #2

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #3

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #4

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #5

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #6

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #7

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

ROUND #8

Please Enter the Follower's ID Number:

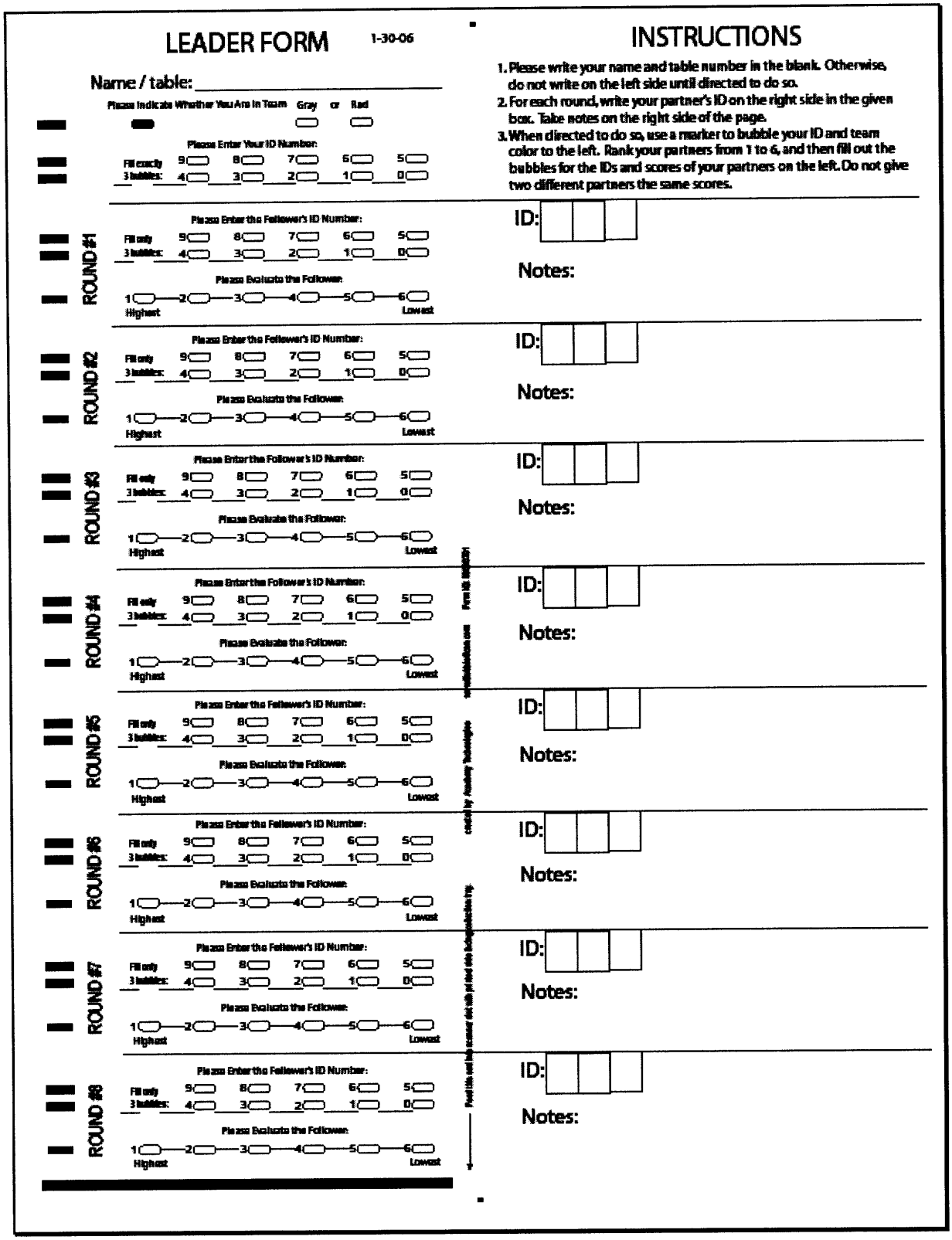
Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 2 3 4 5 6

Lowest Average Highest

A3: Final form used in 2007 Leadership Module.



LEADER FORM 1-30-06

Name / table: _____

Please Indicate Whether You Are In Team Gray or Red

Please Enter Your ID Number:

Fill exactly 3 bubbles: 9 8 7 6 5 4 3 2 1 0

ROUND #1

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #2

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #3

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #4

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #5

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #6

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #7

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

Notes:

ROUND #8

Please Enter the Follower's ID Number:

Fill only 3 bubbles: 9 8 7 6 5 4 3 2 1 0

Please Evaluate the Follower:

1 (Highest) — 2 — 3 — 4 — 5 — 6 (Lowest)

ID:

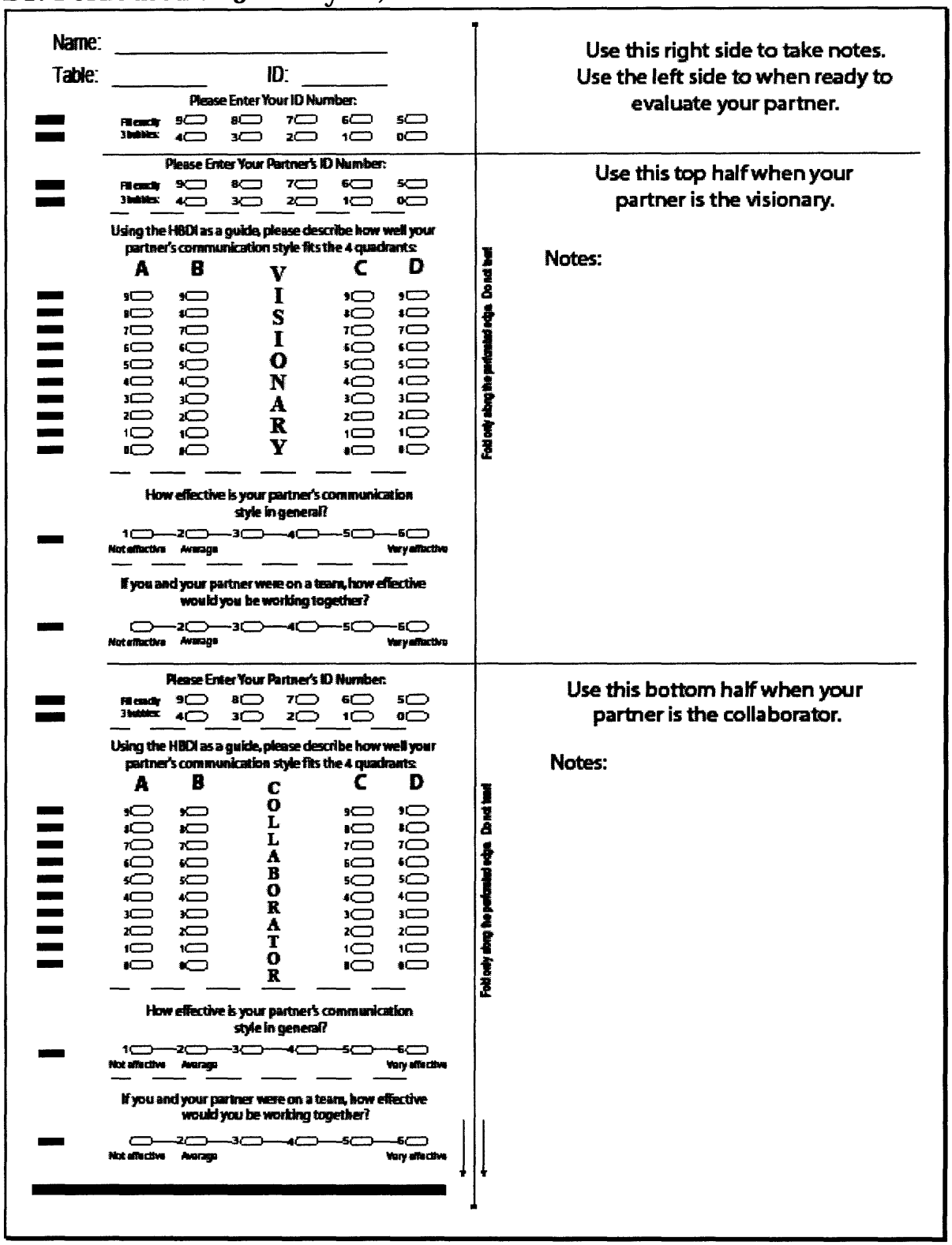
Notes:

INSTRUCTIONS

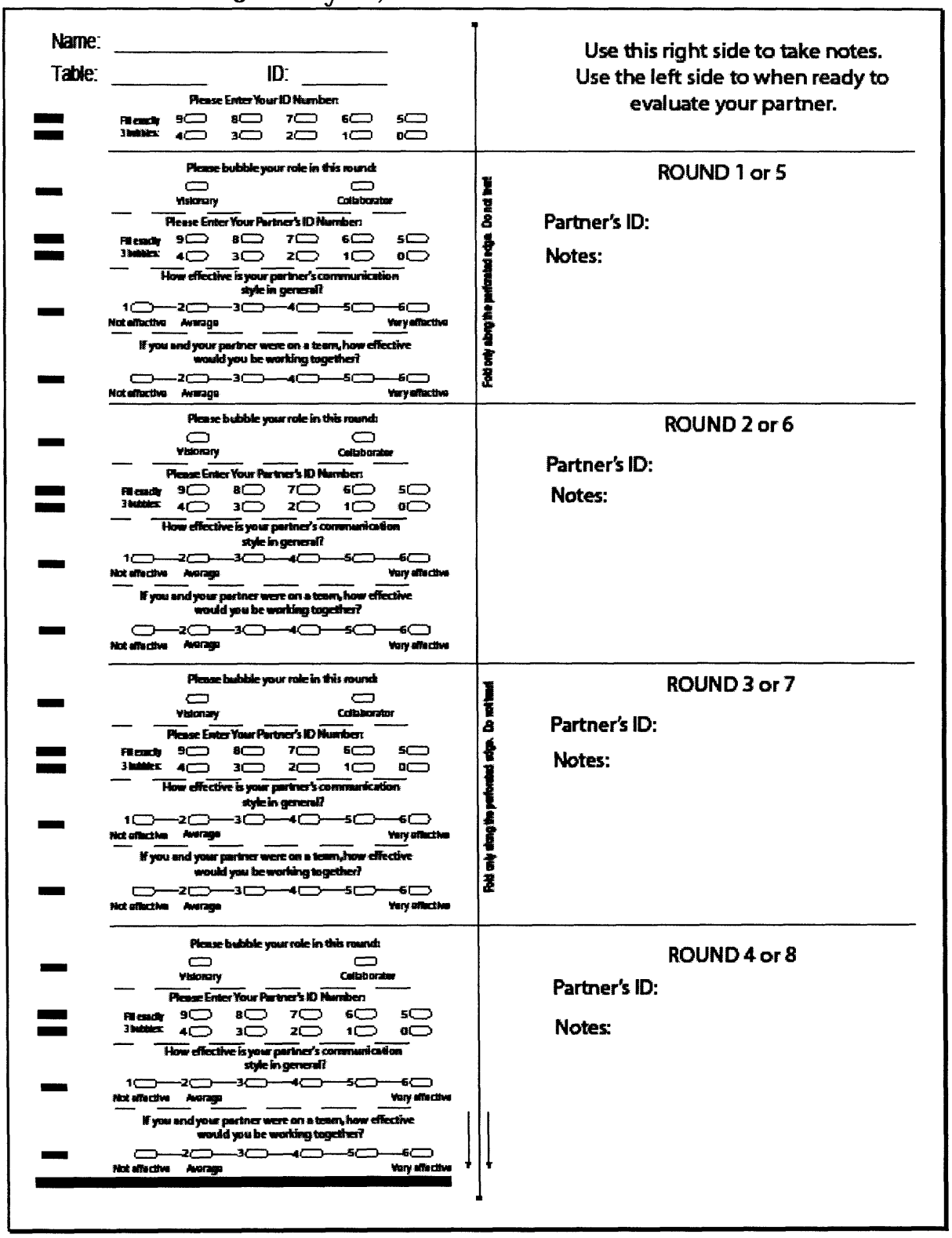
1. Please write your name and table number in the blank. Otherwise, do not write on the left side until directed to do so.
2. For each round, write your partner's ID on the right side in the given box. Take notes on the right side of the page.
3. When directed to do so, use a marker to bubble your ID and team color to the left. Rank your partners from 1 to 6, and then fill out the bubbles for the IDs and scores of your partners on the left. Do not give two different partners the same scores.

Appendix B: Forms for 2008 Leadership Module

B1: Form used on January 16, 2008



B2: Form used on January 30, 2008



Bibliography

[1] interview. (n.d.). *The American Heritage® Dictionary of the English Language, Fourth Edition.* Retrieved March 26, 2008, from Dictionary.com website: http://dictionary.reference.com/browse/interview

[2] Kurzban, Robert and Jason Weeden. "HurryDate: Mate preferences in action." *Evolution and Human Behavior*, Vol. 26 (2005), pg. 227-244.

[3] Ambady, N., & Rosenthal, R. (1993). Half a minute: Predicting teacher evaluations from thin slices of nonverbal behavior and physical attractiveness. *Journal of Personality and Social Psychology, 64*, 431-441.

[4] Prickett, T, Gada-Jain, N, &, Bernieri, F. J. 2000, May, The importance of first impressions in a job interview. Presented at the annual meeting of the Midwestern Psychological Association Convention, Chicago, IL.

[5] Gladwell, Malcolm. "The New Boy Network." *New Yorker*: 29 May 2000. Accessed April 6, 2008. http://www.gladwell.com/2000/2000_05_29_a_interview.htm

[6] Ancona, Deborah, et al. "Leadership in an Age of Uncertainty." Managing for the Future: Organizational Behavior and Processes, 3rd ed. Frederick, MD: Von Hoffmann Press, Inc, 2005.

[7] Eden, Dov, and Uri Leviatan. "Implicit Leadership Theory as a Determinant of the Factor Structure Underlying Supervisory Behavior Scales." *Journal of Applied Psychology*, Vol. 60, No. 6, 1975.

[8] Yukl, Gary. *Leadership in Organizations*, 6th ed. New Jersey: Pearson Prentice Hall, 2006.

[9] Burns, James M. *Leadership.* New York, Harper and Row, 1978.

[10] Bass, Bernard M. *Leadership and Performance Beyond Expectation.* New York, Free Press, 1985.

[11] Wasserman, Stanley and Katherine Faust. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.

[12] Norman, Donald A. "Commentary: Human error and the design of computer systems." Communications of the ACM, 1990, 33, 4-7. Accessed April 21, 2008. http://www.cs.wright.edu/~snarayan/isis/pdf/group11one.pdf

[13] Hermann, Ned. "The Theory Behind the HBDI® and Whole Brain® Technology." HBDI.com. August 1999. Accessed 18 May 2008. http://www.hbdi.com/uploads/100024_articles/100543.pdf

[14] Mussulman, Laura and Deborah White. "The Human Factors Analysis and Classification System (HFACS)." *Approach*, Naval Safety Center, July-August 2004. Accessed April 16, 2008. http://www.safetycenter.navy.mil/MEDIA/approach/issues/julaug04/HumanFactors.htm

[15] Reason, James. "Human Error: Models and Management." *BMJ*, Vol. 320, Pg. 768-770. Accessed April 2, 2008. http://www.bmj.com/cgi/reprint/320/7237/768

[16] Langville, Amy N. and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings.* Princeton University Press, New Jersey, 2006.

[17] "BubbleScan" by *Academy Technologies.* http://www.bubblescan.com/

[18] "OpScan® 6/8 Scanner" by *Pearson Education, Inc.* http://www.pearsonschool.com/index.cfm?locator=PSZ3Rf&PMDbSiteId=2781&PMDbSolutionId=6726&PMDbSubSolutionId=21684&PMDbCategoryId=&PMDbProgramId=33901&level=4

[19] "Clarity 210i and 280i" by *Scantron Corporation.* http://www.scantron.com/scanners/clarity.aspx

[20] "UPOP." http://web.mit.edu/engineering/upop/